DOI:
10.1109/LRA.2018.2853801

Document Version
Peer reviewed version

Reactive Magnetic-field-inspired Navigation Method for Robots in Unknown Convex 3D Environments

Ahmad Ataka[1,2], Hak-Keung Lam[1], and Kaspar Althoefer[2]

***Abstract*—With a shift in current robotics application from known, well-defined environments towards unknown environments, the robot's ability to avoid unknown obstacles in real-time whilst relying on limited information about spatial constraints in its path becomes essential. Taking inspiration from the laws of electromagnetism, we present a novel navigation method, whereby the moving robot induces an artificial electric current onto the obstacle surface generating, in turn, a magnetic field guiding the robot along the obstacle's boundary without affecting its kinetic energy. Our method has several advantages over existing methods: 1) it guides point-like robots towards the goal without suffering from local minima in 3D environments populated with convex obstacles, 2) it does not need any prior knowledge of obstacle positions and geometries, 3) it only requires environmental sensor information that is spatially and temporally local to generate motion commands iteratively. Our navigation method is tested in simulations and experiments, showing that a point-to-point navigation of point-like robots and the end effector of the Baxter's arm has been successfully achieved in a collision-free manner towards a goal position in a 3D environment populated with unknown convex obstacles.**

***Index Terms*—Reactive and Sensor-Based Planning, Collision Avoidance, Motion and Path Planning.**

I. Introduction

THE fields of robot navigation and path planning have been extensively studied in the robotics community over the past thirty years [1], [2]. Developed navigation methods range from classic geometrical planning, which tries to produce a series of collision-free configurations from an initial configuration to the final configuration, to motion planning, which produces the necessary control signals needed to move the robot towards the goal. Achieving collision-free navigation becomes more challenging as the robot applications move from well-defined environments towards unknown environments where the obstacle's position and geometry are not known beforehand. Thus, the robot's ability to reach the goal whilst avoiding obstacles on-line with limited information of the environment becomes important.

Early robot navigation and planning techniques emerged from the well-known piano mover's problem, where the goal is to move a piano through a cluttered environment, often using the configuration space of the robot for planning [3], [1]. Approaches based on the configuration space, though, are plagued by the costly numerical computation, especially when dealing with high-degree-of-freedom robots [4]. Many variants of the original configuration space approach have been developed, with an aim to create practical solutions that can achieve real-time path planning and navigation. For example, sampling-based planning does not require a complete description of the C-Space and can produce planning results more quickly. Popular examples of this approach are probabilistic roadmaps (PRMs) [5] and rapidly-exploring random trees (RRTs) [6] which have triggered a lot of variances as reported in [7]. However, most of these methods assume to have perfect prior knowledge of the environment. Other approaches which aim to generate paths for static environments and modify the path as the environment changes also suffer from the same problem [8], [9].

For a robot with no prior knowledge of the environment, a more suitable method is sensor-based navigation, in which the planning is seen as a control problem. In this approach, the environment is sensed by the robot and the information is used to generate a trajectory on-line by producing a required motion signal considering the dynamics or kinematics model of the robot. It is also possible to use only the current sensor data, hence, forgetting past data, as, for example done in reactive navigation [10]. One popular example of a reactive navigation method is the Artificial Potential Field (APF) method inspired by electrostatic phenomena where the robot is repelled by obstacles and attracted towards the goal [11]. Its simplicity and elegance triggered a lot of research and led to the development of many APF variances such as the one reported in [12]. However, its drawbacks are also well documented, mainly the local minima problem whereby the robot can get stuck in undesired configurations even for fairly simple obstacle configurations even though a solution (a collision-free path to the goal) exists [2].

The inherent local minima in the artificial potential field method motivated the incorporation of a vortex field, where the field produced by an obstacle is designed to circulate the obstacle surface [13]. However, the authors do not provide any stability or collision avoidance guarantee. Taking inspiration from the phenomena of wave expansion, the fast-marching method generates a local-minima-free potential by starting

Manuscript received: February, 24, 2018; Revised May, 22, 2018; Accepted June, 20, 2018.

This paper was recommended for publication by Editor Tamim Asfour upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by King's College London, the EPSRC in the framework of the NCNR (National Centre for Nuclear Robotics) project (EP/R02572X/1), q-bot led project WormBot (2308/104059), and the Indonesia Endowment Fund for Education, Ministry of Finance Republic of Indonesia.

[1]A. Ataka and H.K. Lam are with the Centre for Robotics Research (CoRe), Department of Informatics, King's College London, WC2R 2LS, United Kingdom ahmad_atak_awwalur.rizqi@kcl.ac.uk

[2]A. Ataka and K. Althoefer are with the Centre for Advanced Robotics @ Queen Mary (ARQ), Faculty of Science and Engineering, Queen Mary University of London, Mile End Road, London E1 4NS, United Kingdom k.althoefer@qmul.ac.uk

Digital Object Identifier (DOI): see top of this page.

a "wave" from the goal configuration until it reaches the initial configuration [14]. This method, though, relies on the knowledge of the environment prior to the robot's movement. Another method exploited the power diagram and convex optimization to design a globally stable feedback planner with unique attractor in the goal position [15]. Unlike the classical navigation function method, this method only needs to know the location of the nearby obstacle obtained from an onboard sensor. Nonetheless, this approach is only suitable for a topologically simple planar environment consisting of only spherical obstacles and robot. Other studies suggested applying the properties of an artificially-induced magnetic field for robot navigation, such as [16]-[17], which claim to resolve the problem of entrapment in local minima. This method though also requires the geometry and location of obstacles to be known beforehand. In [18], the concept of circular fields was introduced and applied to partially unknown environments. This method, however, requires prior knowledge of the location of the centre of the obstacles.

The concept of "circular fields", where the field pushes the robot onto a circular path rather than repelling the robot away from the obstacle, as described in [16]-[18], was further extended. Similar approaches under the heading of "gyroscopic force" were used for planar point-like robots to do boundary following [19], obstacle avoidance [20] and, in the context of multi-agent systems, formation control [20]. A recent work explored the possibility of applying the sliding-mode-based navigation to handle maze-like and even dynamic environment [21]. However, all these methods are designed specifically for planar mobile robot applications. There were also recent efforts to apply the concept of gyroscopic forces in 3D for fully-actuated and under-actuated system [22] but the method is limited to cylindrical and spherical obstacles with perfectly-known geometry. Gyroscopic-force-based algorithms were also used for multi-robot formation in 3D [23]-[25]. However, none of these works deals with the problem beyond the scope of collision avoidance among point-like agents for formation control.

In this paper, we present a magnetic-field-inspired algorithm for reactive robot navigation. An artificial electric current, induced by the robot on the obstacle surface, will generate a magnetic field which alters the robot's movement without affecting its energy. The algorithm is able to guide point-like robots towards the goal despite using only local sensory information of the environment and without the need for prior knowledge of obstacle position and geometry. Our algorithm goes beyond the standard APF method, being capable of generating paths in environments with convex obstacle without experiencing local minima. Our approach also outperforms previous magnetic-field-inspired navigation methods [16]-[18] since it does not rely on prior knowledge of the obstacles' geometrical properties. Our approach also provides a clear improvement over gyroscopic force based approaches [19]-[25] as our method is designed to be generic so that it can be used not only for planar mobile robots but also for robots in 3D environments with an unknown arbitrarily shaped convex obstacle. This work extends our previous work which describes the preliminary formulation of the magnetic-field-inspired navigation method applied to mobile robot operating in planar environments only [26]. To the best of our knowledge, this is the first time a general reactive navigation method exploiting magnetic field properties is applied to the problem of robot navigation in 3D environments without prior knowledge of obstacles and capable of guiding the robot to the goal in a globally stable way.

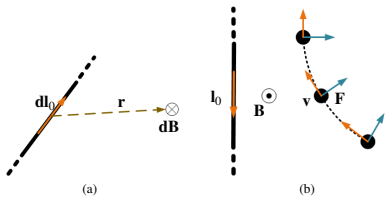


Fig. 1. (a) An electric current flows in the direction of $\mathbf{dl}_o$ will produce a magnetic field **dB**, directed into the page. (b) A charged particle under the influence of the magnetic field **B** produced by current $\mathbf{l}_o$ changes it motion direction due to the force **F**. For robot navigation, this particle can be seen as a robot with the current-carrying wire serves as an obstacle surface.

II. Inspiration

The inspiration for our method comes from the laws of electromagnetism. A wire segment of length $\mathbf{dl}_o$ carrying an electrical current i_o as depicted in Fig. 1a will produce a magnetic field **dB** as described in the following [27]

$$\mathbf{dB} = \frac{\mu_0}{4\pi}\frac{i_o \mathbf{dl}_o \times \mathbf{r}}{|\mathbf{r}|^3}, \tag{1}$$

where $\mathbf{r}$ denotes the position vector of a point in space with respect to the wire segment, μ_0 is a permeability constant, and $\times$ denotes the vector cross product operation.

A particle with positive charge q moving close to the current-carrying wire will be affected by the presence of the magnetic field as depicted in Fig. 1b. A force **F** whose direction is perpendicular to both particle's velocity $\mathbf{v}$ and magnetic field **B** will be applied to the particle as a result of this interaction and is given by [27]

$$\mathbf{F} = q\mathbf{v} \times \mathbf{B}. \tag{2}$$

Substituting **B** from eq. (1) and dropping the infinitesimal notation, the force acting on the particle is given by

$$\mathbf{F} = \frac{\mu_0 q i_o}{4\pi} \frac{\mathbf{v} \times (\mathbf{l}_o \times \mathbf{r})}{|\mathbf{r}|^3}. \tag{3}$$

The magnetic field interacts with the electric charge in a way that the produced force is always perpendicular to the moving charge velocity direction and, thus, changing its movement direction. Inspired by this physical phenomenon, we can think of a robot as a charged particle with velocity $\mathbf{v}$ and the obstacle surface as a current-carrying wire. The artificial current $\mathbf{l}_o$ flowing on the obstacle surface located at position $\mathbf{r}_o$ with respect to the robot is designed in such a way that generated force **F** will guide the robot away from a head-on collision and follow obstacle boundary instead.

It is noted that, by definition, $\mathbf{r}_o = -\mathbf{r}$, so eq. (3) can be rewritten in a more general form as follows

$$\mathbf{F} = c\ \mathbf{l}_a \times (\mathbf{r}_o \times \mathbf{l}_o)\ \ f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|), \tag{4}$$

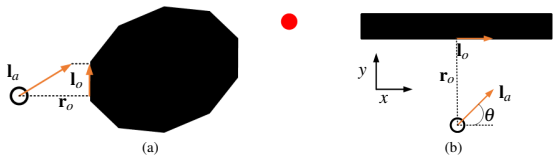


Fig. 2. (a) The workspace of a point-like robot moves towards the goal (red circle) in the vicinity of the polygonal obstacle (black). $\mathbf{l}_o$ and $\mathbf{l}_a$ denote the artificial current on the obstacle surface and the robot's velocity. (b) The scenario of a robot moving in the vicinity of a flat obstacle surface.

where $c > 0$ is a scalar constant, $\mathbf{l}_a$ stands for the robot's velocity direction, $\mathbf{l}_o$ stands for the artificial current on the obstacle surface, and $f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|) \geq 0$ is a scalar function which depends on the robot-to-obstacle distance $|\mathbf{r}_o|$ and/or robot's speed $|\dot{\mathbf{p}}|$. We introduce a skew-symmetric matrix $\hat{\mathbf{l}}$ to replace the vector cross product operation $\mathbf{l}\times$ of a vector $\mathbf{l} = \begin{bmatrix} l_x & l_y & l_z \end{bmatrix}^T$ as follows

$$\hat{\mathbf{l}} = \begin{bmatrix} 0 & -l_z & l_y \\ l_z & 0 & -l_x \\ -l_y & l_x & 0 \end{bmatrix}. \tag{5}$$

III. Proposed Algorithm

To explain the idea of artificial current generation on the obstacle surface, it is better to analyze a point-mass robot moving in $\mathbb{R}^2$, as depicted in Fig 2a. When the robot first senses the obstacle surface in front of it, there are 2 choices of movement: either following the surface to the left or right. To minimize unwanted oscillation, the induced current direction $\mathbf{l}_o$ on the obstacle surface (which will be followed by the robot) is designed to be the projection of the robot's velocity direction $\mathbf{l}_a$ on to the obstacle surface.

With no knowledge of the obstacle geometry, we can create a reasonable assumption regarding the obstacle surface relying only on range sensor information. For an obstacle whose closest point sensed by the robot located in position $\mathbf{r}_o$ with respect to the robot, we assume that the obstacle surface at this point is perpendicular to the direction of $\mathbf{r}_o$. This simplification is reasonable since we want the robot to move perpendicular to the vector connecting the robot and the closest obstacle point at any time. Using geometry, the current direction $\mathbf{l}_o$ can then be written as

$$\mathbf{l}_o = \mathbf{l}_a - \frac{(\mathbf{l}_a^T \mathbf{r}_o)\mathbf{r}_o}{|\mathbf{r}_o|^2}. \tag{6}$$

Using this formulation, there could be a special case where the artificial current becomes zero, i.e. when the robot's direction $\mathbf{l}_a$ is in line with vector $\mathbf{r}_o$. The solution to this problem will be explained at the end of this section, where the current will be slightly modified into (29).

To make the robot move towards the direction of $\mathbf{l}_o$, the force equation in (4) is modified as follows

$$\mathbf{F} = c \;\; \mathbf{l}_a \times (\mathbf{l}_o \times \mathbf{l}_a) \;\; f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|). \tag{7}$$

For a robot with position vector $\mathbf{p}$, the robot's velocity direction $\mathbf{l}_a$ is defined as $\mathbf{l}_a = \frac{\dot{\mathbf{p}}}{|\dot{\mathbf{p}}|}$.

Lemma 1. *The obstacle avoidance force* $\mathbf{F}$ *in (7) will not change the magnitude of the robot's velocity* $|\dot{\mathbf{p}}|$.

Proof. Assuming $cf(|\mathbf{r}_o|, |\dot{\mathbf{p}}|) = 1$ and $\mathbf{l}_o \times \mathbf{l}_a = \mathbf{a}$, using the skew-symmetric definition in (5), we get

$$\mathbf{l}_a^T \mathbf{F} = \mathbf{l}_a^T (\hat{\mathbf{l}}_a \mathbf{a}) = 0. \tag{8}$$

Using Newton's law of motion, for a robot with mass m, speed $v = |\dot{\mathbf{p}}|$, and velocity $\dot{\mathbf{p}} = v\mathbf{l}_a$, we get

$$\mathbf{F} = m\frac{d\dot{\mathbf{p}}}{dt} = m(\frac{dv}{dt}\mathbf{l}_a + v\frac{d\mathbf{l}_a}{dt}). \tag{9}$$

Combining with (8), we get

$$\mathbf{l}_a^T \mathbf{F} = m(\frac{dv}{dt} + v\mathbf{l}_a^T \frac{d\mathbf{l}_a}{dt}) = 0. \tag{10}$$

Due to the fact that $\mathbf{l}_a$ is a unit vector with a constant magnitude $|\mathbf{l}_a| = 1$, $\frac{d\mathbf{l}_a}{dt}$ refers to the change of the vector's direction only. From geometry, the change in a vector's direction is perpendicular to the vector direction itself. Hence, we can conclude that $\mathbf{l}_a^T \frac{d\mathbf{l}_a}{dt} = 0$, and then we can simplify (10) into $\mathbf{l}_a^T \mathbf{F} = m\frac{dv}{dt} = 0$. Since the mass is not zero, the only solution is $\frac{dv}{dt} = 0$, i.e. the speed is constant. □

Lemma 2. *The force* $\mathbf{F}$ *whose direction is described in (7) created by induced current* $\mathbf{l}_o$ *of the obstacle will tend to move the robot in the direction of induced current* $\mathbf{l}_o$.

Proof. Suppose we define $d = \frac{(\mathbf{l}_a^T \mathbf{r}_o)}{|\mathbf{r}_o|^2}$ and $cf(|\mathbf{r}_o|, |\dot{\mathbf{p}}|) = 1$, using definition in (6) and the fact that $\hat{\mathbf{l}}_a \mathbf{l}_a = \mathbf{0}$, the force $\mathbf{F}$ will be $\mathbf{F} = d(\mathbf{r}_o \times \mathbf{l}_a) \times \mathbf{l}_a$. The product of this force $\mathbf{F}$ and obstacle current $\mathbf{l}_o$ can be written as

$$\mathbf{F}^T \mathbf{l}_o = d((\mathbf{r}_o \times \mathbf{l}_a) \times \mathbf{l}_a)^T (\mathbf{l}_a - d\mathbf{r}_o) = g - h, \tag{11}$$

where $g = d((\mathbf{r}_o \times \mathbf{l}_a) \times \mathbf{l}_a)^T \mathbf{l}_a$ and $h = d^2((\mathbf{r}_o \times \mathbf{l}_a) \times \mathbf{l}_a)^T \mathbf{r}_o$. The scalar g can be simplified as

$$g = -d\mathbf{l}_a^T (\mathbf{l}_a \times (\mathbf{r}_o \times \mathbf{l}_a)) = -d\mathbf{l}_a^T (\hat{\mathbf{l}}_a \hat{\mathbf{r}}_o \mathbf{l}_a)). \tag{12}$$

By definition of the skew-symmetric matrix in (5), we can see that for any vector $\mathbf{a}$, the term $\mathbf{l}_a^T (\hat{\mathbf{l}}_a \mathbf{a}) = 0$, so we can conclude that $g = 0$. To simplify the term h, without losing generality, we can write down $\mathbf{l}_a$ and $\mathbf{r}_o$ as $\mathbf{l}_a = \begin{bmatrix} l_{ax} & l_{ay} \end{bmatrix}^T$ and $\mathbf{r}_o = \begin{bmatrix} r_{ox} & r_{oy} \end{bmatrix}^T$. This will lead to

$$h = -d^2(-l_{ay} r_{ax} + l_{ax} r_{ay})^2 \leq 0, \tag{13}$$

$$\mathbf{F}^T \mathbf{l}_o = d^2(-l_{ay} r_{ax} + l_{ax} r_{ay})^2 \geq 0. \tag{14}$$

This concludes that the component of force $\mathbf{F}$ in the direction of $\mathbf{l}_o$ will always have the same direction of $\mathbf{l}_o$, i.e. tend to move the robot towards the direction of $\mathbf{l}_o$. □

Lemma 3. *Assuming the obstacle surface to be flat and the robot does not collide with an obstacle, the final direction of the robot due to the force* $\mathbf{F}$ *whose direction described in (7) will be parallel to the obstacle surface.*

Proof. Without loss of generality, we assume that the obstacle surface lies on x-plane as depicted in Fig. 2b, so the closest point on the obstacle is located at $\mathbf{r}_o = \begin{bmatrix} 0 & r_o \end{bmatrix}^T$ with respect

to the robot. Defining angle between $\mathbf{l}_a$ and $\mathbf{l}_o$ as θ, we write down the robot's velocity direction as $\mathbf{l}_a = \begin{bmatrix}\cos\theta & \sin\theta\end{bmatrix}^T$. From (6), the obstacle current is given by $\mathbf{l}_o = \begin{bmatrix}\cos\theta & 0\end{bmatrix}^T$. Assuming $f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|) = 1$, the force $\mathbf{F}$ from (7) will be

$$\mathbf{F} = c\hat{\mathbf{l}}_a(\hat{\mathbf{l}}_o\mathbf{l}_a) = c\begin{bmatrix}\sin^2\theta\cos\theta & -\sin\theta\cos^2\theta\end{bmatrix}^T. \quad (15)$$

From Lemma 1, a robot with a constant speed v will have the following equation of motion

$$\mathbf{F} = mv\frac{d\mathbf{l}_a}{dt} = mv\dot{\theta}\begin{bmatrix}-\sin\theta & \cos\theta\end{bmatrix}^T. \quad (16)$$

Without loss of generality, suppose we analyze the equation of motion in y direction and combine (15) and (16) to obtain the equation as follows

$$mv\frac{d\theta}{dt}\cos\theta = -c\sin\theta\cos^2\theta. \quad (17)$$

After simplification, we will get the following equation

$$\int \frac{1}{\cos\theta\sin\theta}d\theta = -\int \frac{c}{mv}dt. \quad (18)$$

The final equation will have the following form

$$\theta(t) = \arctan(Ae^{-\frac{c}{mv}t}), \quad (19)$$

where A is a constant which depends on the initial angle when the robot senses the obstacle for the first time. As we can see, as $t \to \infty$, $\theta \to 0$, meaning that the robot's final direction is parallel to the robot surface. □

To complete the description of the algorithm, we choose the function $f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|)$ to have the following form

$$f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|) = \frac{|\dot{\mathbf{p}}|}{|\mathbf{r}_o|}. \quad (20)$$

The reason for this is that we want the force to be larger when the robot comes closer to the obstacle and when the robot has a higher speed.

Lemma 4. *For a robot with velocity direction $\mathbf{l}_a$ located at initial distance $|\mathbf{r}_o|$ from the flat obstacle with induced current $\mathbf{l}_o$ described in (6), the robot's distance to the obstacle surface will never be zero when the initial direction of $\mathbf{l}_a$ is not in the direction of $\mathbf{r}_o$.*

Proof. Since we do not use assumption $f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|) = 1$ any longer, eq. (17) can be rewritten as

$$mv\frac{d\theta}{dt}\cos\theta = -c\sin\theta\cos^2\theta\frac{v}{r} \Leftrightarrow \dot{\theta} = -\frac{c}{mr}\cos\theta\sin\theta, \quad (21)$$

where r stands for the distance to the obstacle surface. For the scenario in Fig. 2b, the velocity component in y direction is the same as $-\dot{r}$ as follows

$$\dot{r} = -v\sin\theta \Leftrightarrow \frac{dr}{d\theta}\dot{\theta} = -v\sin\theta. \quad (22)$$

Substituting (21) into (22) and simplifying, we obtain

$$\int \frac{1}{r}dr = \frac{mv}{c}\int \frac{1}{\cos\theta}d\theta. \quad (23)$$

The final equations has the following form

$$r(\theta) = B|\sec\theta + \tan\theta|^C, \quad (24)$$

$$B = \frac{r_0}{|\sec\theta_0 + \tan\theta_0|^C}, \quad (25)$$

where r_0 stands for initial distance to the obstacle when the angle between the robot's velocity to the obstacle surface is θ_0 and $C = \frac{mv}{c}$. The closest distance between the robot and obstacle occurs when $\dot{\mathbf{r}} = -v\sin\theta = 0$, i.e. $\theta = 0$. Substituting the value $\theta = 0$ to (24), we get that the robot's closest distance to the obstacle is given by $r_f = B$. As we can see, the value of B never reaches zero except when $\theta_0 = \pi/2$, i.e. when the robot's initial direction $\mathbf{l}_a$ is parallel to $\mathbf{r}_o$. □

Remark 1. *Lemma 4 guarantees that the robot's distance to the flat obstacle surface will never be zero. Hence, by definition of the convex set, this property will also be true for any convex-shaped obstacle.*

Lemma 5. *The final equilibrium direction of the robot described in Lemma 3 is globally asymptotically stable.*

Proof. Suppose we define the Lyapunov function candidate as follows

$$V = -\ln(\mathbf{l}_a^T\mathbf{l}_o). \quad (26)$$

This form is chosen since it reflects how much the direction of the robot $\mathbf{l}_a$ deviates from the obstacle surface $\mathbf{l}_o$. For this Lyapunov function candidate, we get

$$\dot{V} = -\frac{\dot{\mathbf{l}}_a^T\mathbf{l}_o + \mathbf{l}_a^T\dot{\mathbf{l}}_o}{\mathbf{l}_a^T\mathbf{l}_o}. \quad (27)$$

Using the scenario in Fig. 2b, we can write down the rate of each vector as $\dot{\mathbf{l}}_a = \dot{\theta}\begin{bmatrix}-\sin\theta & \cos\theta\end{bmatrix}^T$ and $\dot{\mathbf{l}}_o = \dot{\theta}\begin{bmatrix}-\sin\theta & 0\end{bmatrix}^T$. Substituting (21) into the equation, we get

$$\dot{V} = -\frac{2c}{mr}\sin^2\theta. \quad (28)$$

By definition, both c and m are always positive, while, according to Lemma 4, except at special condition $\theta_0 = \frac{\pi}{2}$, r will always be greater than zero. Hence, $\dot{V}$ will always be negative except when $\theta = 0$ or $\theta = \pi$, which are the equilibrium points in which $V = 0$ and $\dot{V} = 0$. Both cases happen when the robot is parallel to the obstacle surface and hence, conclude the proof. □

Lemma 4 shows that the robot will never touch the obstacle surface except when the initial robot direction $\mathbf{l}_a$ is parallel to the obstacle position with respect to the robot $\mathbf{r}_o$. In reality, this special condition is not likely to happen, as the imperfection of sensor or actuator will make the vector $\mathbf{l}_a$ and $\mathbf{r}_o$ still have an angle between them albeit small. However, to avoid this problem, when the current magnitude on the obstacle surface is smaller than some small positive constant ε, we take the unit vector as the obstacle current. The current generation in (6) can be modified as follows

$$\mathbf{l}_o = \begin{cases}\mathbf{l}_{o,i} & \text{if } |\mathbf{l}_{o,i}| > \varepsilon \\ \dfrac{\mathbf{l}_{o,i}}{|\mathbf{l}_{o,i}|} & \text{if } |\mathbf{l}_{o,i}| \le \varepsilon\end{cases}, \quad (29)$$

where $\mathbf{l}_{o,i}$ is obstacle current in (6).

IV. Implementation

A point-like robot has a dynamic model as follows

$$\ddot{\mathbf{p}} = \mathbf{u} \tag{30}$$

where $\mathbf{p} \in \mathbb{R}^3$ stands for the robot's position and $\mathbf{u} \in \mathbb{R}^3$ stands for the acceleration input. The control law which will guide the robot towards the goal and avoid obstacles is

$$\mathbf{u} = \mathbf{F_g} + \mathbf{F_o}. \tag{31}$$

$\mathbf{F_g}$ is an attractive PD control towards a goal $\mathbf{p_g}$ given by

$$\mathbf{F_g} = -K_P(\mathbf{p} - \mathbf{p_g}) - K_D\dot{\mathbf{p}}, \tag{32}$$

where K_P and K_D are positive constants. The obstacle avoidance terms $\mathbf{F_o}$ is a magnetic force described from (7), (20), and (29) and will be activated once the distance between robot and obstacle closer than a limit distance r_l as follows

$$\mathbf{F_o} = \begin{cases} c\ \mathbf{l}_a \times (\mathbf{l}_o \times \mathbf{l}_a)\ \frac{|\dot{\mathbf{p}}|}{|\mathbf{r}_o|} & \text{if } |\mathbf{r}_o| < r_l \\ \mathbf{0} & \text{if } |\mathbf{r}_o| \geq r_l \end{cases}. \tag{33}$$

The robot in this case is assumed to have a 360° laser sensor so that it can sense the surrounding environments as far as distance r_l in all directions from the robot.

With the property of the field explained in Section III, for a non-saturated input, it is shown that the robot will not collide with an obstacle (Lemma 4). Assuming this is the case, we can show the stability of control law in (31).

Lemma 6. *The control law in (31)-(33) for a point-like robot modeled in (30) is globally asymptotically stable.*

Proof. Suppose we have the Lyapunov function candidate as follows

$$V = \frac{1}{2}\dot{\mathbf{p}}^T\dot{\mathbf{p}} + \frac{1}{2}K_P\mathbf{e}^T\mathbf{e}, \tag{34}$$

where $\mathbf{e} = (\mathbf{p} - \mathbf{p_g})$ represents error vector. Differentiating with respect to time, we get

$$\dot{V} = \dot{\mathbf{p}}^T\ddot{\mathbf{p}} + K_P\mathbf{e}^T\dot{\mathbf{e}}. \tag{35}$$

Substituting (31)-(33) and the fact that $\dot{\mathbf{e}} = \dot{\mathbf{p}}$ and $\dot{\mathbf{p}}^T\mathbf{F_o} = 0$ according to Lemma 1, we get $\dot{V} = -K_D\dot{\mathbf{p}}^T\dot{\mathbf{p}}$. We can see that $\dot{V}$ is always negative except at equilibrium state $\mathbf{p} = \mathbf{p_g}$ and $\dot{\mathbf{p}} = \mathbf{0}$ where $V = \dot{V} = 0$, and hence, this equilibrium is globally asymptotically stable. □

It is noted that when the obstacle is so large that it is possible for the robot to be too far from the goal along the obstacle's boundary, the goal attraction component of our algorithm could decrease the robot's speed to zero. At this point, the obstacle avoidance term will also be zero, and the attraction to the goal could then force the robot to reverse its travel direction, which could result in a condition where the robot will oscillate without being able to reach the goal.

To avoid this problem, we add a goal relaxation (GR) term which will decrease the contribution of the goal attraction as the robot follows the boundary of the obstacle, reducing the goal attraction when the robot gets closer to the obstacle as described by

$$w_1 = (1 - e^{-\frac{|\mathbf{r}_o|}{\alpha r_l}}), \tag{36}$$

TABLE I
List of Parameter Values

Param	Point-Like Robot	Baxter
K_P	0.1	10
K_D	0.5	10
c	5	20
r_l	3 m	0.1 m
ε	0.05	0.05
α	1.0	1.0
υ	0.1	0.1

where α is a positive constant. The second property of the GR term is that it will get smaller if the goal is still occluded by an obstacle and will get bigger if the obstacle does not obstruct the goal any longer as described by

$$w_2 = \begin{cases} 1 - \frac{\mathbf{r}_g^T\mathbf{r}_o}{|\mathbf{r}_g||\mathbf{r}_o|} & \text{if } |\mathbf{r}_o| < r_l \\ 1 & \text{if } |\mathbf{r}_o| \geq r_l \end{cases}. \tag{37}$$

For a large obstacle, the attractive term should be even smaller when the robot reaches a point where the distance to the goal $|\mathbf{r}_g|$ is larger than limit value r_{gl} whilst, at the same time, being in close proximity of the obstacle. One way to limit distance r_{gl} is using the initial distance to the goal r_{gi}. The weight is expressed as

$$w_3 = \begin{cases} e^{-\frac{|\mathbf{r}_g| - r_{gl}}{\upsilon}} & \text{if } |\mathbf{r}_o| < r_l \text{ and } |\mathbf{r}_g| \geq r_{gl} \\ 1 & \text{otherwise} \end{cases}, \tag{38}$$

where υ is a positive constant. The overall GR term is then expressed as $\gamma(\mathbf{r}_o, \mathbf{r}_g) = w_1w_2w_3$, and the overall control signal in (31) will be modified as follows

$$\mathbf{u} = \gamma(\mathbf{r}_o, \mathbf{r}_g)\mathbf{F_g} + \mathbf{F_o}. \tag{39}$$

V. Results and Analysis

The magnetic-field-inspired navigation algorithm is implemented to guide a point-like robot model in several simulation scenarios. To better demonstrate the performance of the algorithm, we also implement the algorithm to do point-to-point navigation of the tip of a 7 DOFs Baxter arm towards the goal. It is assumed that the robot only knows the surrounding environment as far as r_l in all directions from the robot's position. We compare the performance of the proposed magnetic-field-inspired (MFI) algorithm with several reactive algorithms (particularly the APF [11], variable speed force field (VSFF) [12], circular field (CF) [18], and gyroscopic force (GF) method [24], [25]). These reactive methods are chosen since they do not rely on the availability of the environmental map, do not need the information regarding the obstacle's shape or geometry, and can be applied in a 3D environment with arbitrary-shaped obstacles. We also compare the performance of the MFI algorithm to the other algorithms when the goal relaxation (GR) term is added, except for the APF method since it can be seen theoretically that the GR terms will not stop the APF in avoiding the local minima. All simulations and experiments are conducted in the Robot Operating System (ROS) framework [28]. The constants used

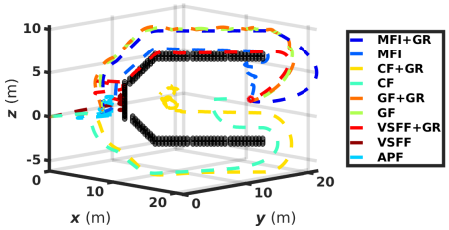


Fig. 3. The trajectory of the robot is drawn in dashed lines moving towards the goal (red mark) with a U-shaped obstacle drawn in black.

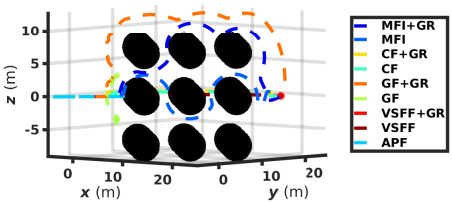
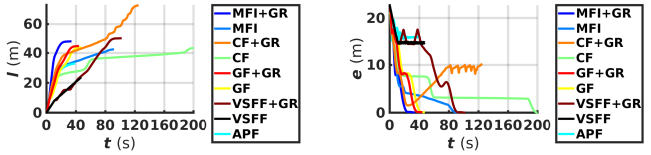
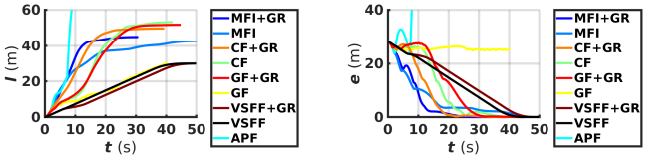


Fig. 5. The trajectory of the robot is drawn in dashed lines moving towards the goal (red mark) and avoiding multiple spherical obstacles drawn in black.



(a) (b)

Fig. 4. The plot of (a) the trajectory covered by the robot $l(t)$ and (b) the position error $e(t)$ for a U-shaped obstacle.



(a) (b)

Fig. 6. The plot of (a) the robot trajectory and (b) the position error for multiple spherical obstacles.

in simulations and experiments, retrieved from trial and error, are listed in Table I. These parameters can be easily tuned without radically disturbing the performance of the algorithm since they only influence how strong the goal attraction is and how close the robot will be to the obstacle surface. Simulation and experimental results are shown in the video attachment.

A. Simulation Results

Fig. 3 shows the trajectory of the robot for an environment consisting of a U-shaped obstacle. We can see that of all navigation algorithms, the APF and VSFF fail to navigate the robot passed the obstacle as the robot gets stuck in a local minimum, while the rest of the algorithms without the goal relaxation successfully reach the goal. This is due to the repulsive nature of the field produced by the obstacle used in the APF and VSFF methods. When we add the algorithms with the goal relaxation, the CF method fails to guide the robot towards the goal due to the fact that the GR term decreases the influence of the goal attraction, failing to attract the robot out of the obstacle influence. This is not the case for the MFI algorithm since the artificial current depends on the robot's speed, and hence, the obstacle influence automatically decreases when the robot slows down.

Among the 6 methods successfully navigating the robot towards the goal, the MFI without the GR has the shortest trajectory as shown in Fig. 4a. However, it requires the third longest time to reach the goal (Fig. 4b) because the robot is still being influenced by the obstacle, despite the fact that the remaining path to the goal is free from obstacles. The same problem occurs for CF which takes the longest time to reach the goal (Fig. 4b). Here, with the introduction of the GR term, as depicted in Fig. 4b, the MFI algorithm reaches the goal in the shortest time when compared to the rest of the algorithms, even when others also use the GR term. The GF algorithm with GR, even though converge slightly faster compared to when the GR term is not used, still converge in a slower time to the MFI due to the repulsive field it employs to keep the robot at a safe distance from the obstacle surface, which could decrease the robot's speed.

Fig. 5 shows the trajectory of the robot for an environment consisting of multiple spherical obstacles. From Fig. 6b, we can see that of all navigation algorithms, the APF and GF fail to navigate the robot passed the obstacle while the rest of the algorithms without the goal relaxation successfully reach the goal. For the case of GF, the robot falls into a zero-speed problem, where the robot's speed decreases due to the goal attraction in following the obstacle boundary, to the point where its speed becomes zero, causing it to move back and forth without being able to reach the goal.

Among the other methods which successfully navigate the robot towards the goal, the MFI with and without the GR converge in a short trajectory as shown in Fig. 6a, only second to the VSFF with and without GR. In terms of the convergence time, as depicted in Fig. 6b, the MFI algorithm reaches the goal in the shortest time when compared to the rest of the algorithms. The CF algorithm with GR also converges in a very short time; however, unlike the MFI, it still relies on the information of the obstacle's central position in reaching this performance.

The scenario in which the GR is even more crucial is one where large obstacle areas are encountered as shown in Fig. 7. This environment consists of an obstacle so large that, in order to circulate around the obstacle, the robot needs to go through a critical point whose distance to the goal is farther than the distance between the goal and the robot's start point.

The problem of local minima occurs for the APF, VSFF without GR, and even VSFF with GR, while the problem of zero speed occurs for the case of the MFI, CF, and GF methods without GR as shown in Fig. 7 and Fig. 8. In this case, the attraction to the goal causes the robot's speed to decrease while following the obstacle's boundary until it becomes zero. Added with the GR, the MFI, CF, and GF successfully avoid

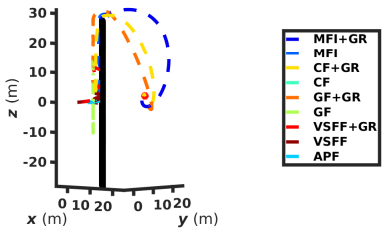
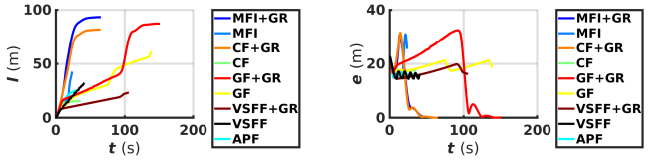


Fig. 7. The robot trajectory is drawn in dashed lines moving towards the goal (red mark) with long planar obstacle drawn in black.



(a) (b)

Fig. 8. The plot of (a) the trajectory covered by the robot $l(t)$ and (b) the position error $e(t)$ as a function of time for the case of large obstacles.

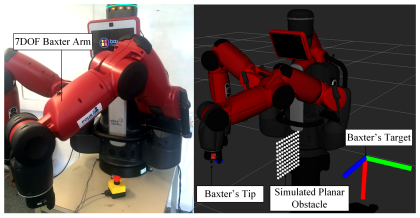


Fig. 9. The experimental setup is shown in the left. The tip of 7 DOFs Baxter arm is used as a point-like robot in 3D. The configuration of the robot, planar obstacle, and desired tip position are shown in the right.

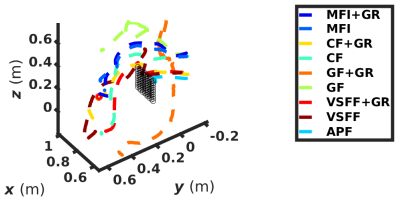


Fig. 10. The trajectory of the Baxter tip is drawn in dashed lines moving towards the goal with obstacles drawn in black.

the obstacle and reach the goal. Among these 3 methods, the MFI and CF have the fastest convergence time due to the fact that the repulsive term in GF slows the robot down (see Fig. 8b). However, it is noted that the MFI method achieves this performance without any information of the environment, unlike the CF which still needs the position of the obstacle's centre point.

B. Experimental Results

Next, we implement our algorithm to guide the tip of a 7 DOFs Baxter Arm (see Fig. 9). Here, the Baxter is opertated using the joint torque control mode in which the joint torque commands that we send is applied in addition to the gravity compensation term and internal joint spring compensation. The control signal described in (31), (39) is applied at the tip as a task-space force, from which a joint torque is computed via Jacobian transpose relation, and the joint acceleration is computed using inverse dynamics.

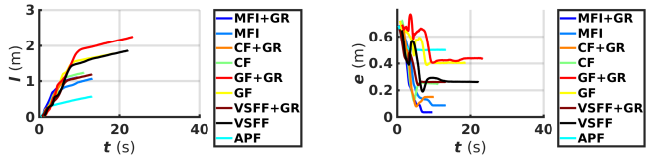
In Fig. 10, we can see that the APF is the only method which fails to guide the robot's tip pass the obstacle. Besides, Fig. 11b shows that the GF and VSFF methods fail to perfectly guide the tip to reach the goal due to their repulsive term causing the tip to be repelled too far and the Baxter arm stuck in its joint constraint. In Fig. 11a, we can observe that the final covered distance for the MFI with GR and without GR are both smaller compared to the other methods. Fig. 11b shows that the MFI and the CF, both with GR and without GR, have the fastest error convergence rate compared to the rest of the methods, shown as a very steep negative gradient at the beginning of the movement. Once again, we argue that the MFI still outperforms the CF due to the fact that it is able to achieve a comparable performance in terms of the convergence time to the CF even with less information.

Table II summarizes the results, comparing the APF, CF, GF, and MFI methods both with GR and without GR. We compare the ability of these methods to successfully reach the target and the path's quality, characterized by the final path $l(t)$ and the time needed for the distance error $e(t)$ to drop below a specified value e_b. Both variables are undefined when the algorithm fails to guide the robot to reach the goal. The value of e_b is chosen to be $e_b = 0.05 r_{gi}$ for the simulations, where r_{gi} stands for initial distance to the goal and $e_b = 0.2$ m for the experimental study due to the possible steady-state positional error of the Baxter's tip.

Table II demonstrates how in both aspects, simulation and experiments, our reactive magnetic-inspired-field navigation method outperforms the other methods in almost every scenario, especially in terms of its ability to guide the robot successfully to the goal and its low error convergence time. We can see that the only algorithm which always works in every scenario is the MFI with GR while the one without GR only fails for the case of the large obstacle. The MFI without GR, in every scenario except the multiple spheres always produce the shortest trajectory. Despite taking longer trajectory, the MFI with GR always has the best performance, or the second best after the CF with GR for some cases, in terms of convergence time despite only requires knowledge of the distance to the closest obstacle and the robot's speed.

VI. Conclusion

We present a reactive magnetic-field-inspired navigation method capable of navigating the robot towards the desired position. The algorithm takes inspiration from the magnetic field laws and is used to guide the movements of a robot, represented by a charged particle. The algorithm outperforms the artificial potential field (APF) method because it is free from local minima in convex environments. The algorithm also improves on earlier magnetic-field-inspired navigation methods found in the literature, as it does not rely on a priori information of the obstacles' geometries and locations. The algorithm has been implemented successfully for a point-like



(a) (b)

Fig. 11. The plot of (a) the trajectory covered by Baxter's tip $l(t)$ and (b) the position error $e(t)$ as a function of time for the case of flat-surfaced obstacles.

TABLE II
SUMMARY OF RESULTS

Robot	Obstacle	Algorithm	Success	Covered Path (m)	Time (s)
Simulation	U-Shaped	APF	×	-	-
		VSFF	×	-	-
		VSFF+GR	✓	49.84	87.28
		GF	✓	43.67	38.18
		GF+GR	✓	44.62	33.23
		CF	✓	43.46	192.88
		CF+GR	×	-	-
		MFI	✓	42.38	82.00
		MFI+GR	✓	47.77	20.28
	Multiple Spheres	APF	×	-	-
		VSFF	✓	30.03	38.57
		VSFF+GR	✓	30.02	42.01
		GF	×	-	-
		GF+GR	✓	51.68	28.50
		CF	✓	53.13	24.80
		CF+GR	✓	49.59	18.70
		MFI	✓	42.66	40.01
		MFI+GR	✓	44.88	19.08
	Long Plane	APF	×	-	-
		VSFF	×	-	-
		VSFF+GR	×	-	-
		GF	×	-	-
		GF+GR	✓	87.25	122.17
		CF	×	-	-
		CF+GR	✓	81.72	39.68
		MFI	×	-	-
		MFI+GR	✓	93.25	40.45
Experiment	Plane	APF	×	-	-
		VSFF	✓	1.86	6.74
		VSFF+GR	×	-	-
		GF	×	-	-
		GF+GR	×	-	-
		CF	✓	1.24	3.95
		CF+GR	✓	1.11	4.00
		MFI	✓	1.07	5.10
		MFI+GR	✓	1.11	3.87

robot in $\mathbb{R}^3$, and also in an experimental setup using a 7DOF Baxter arm. The results show that the algorithm outperforms other reactive navigations. Our algorithm is also a promising candidate to be used for the navigation of other type of robots, such as flying robots, continuum manipulators, and swarming multi-robot systems. Efforts to improve the algorithm by considering the saturation of the robot's actuators, uncertainty in the sensor measurement, the orientation of the robot, and environments with non-convex obstacles will be taken into account in the future.

REFERENCES

[1] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.
[2] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA: MIT Press, Jun. 2005.
[3] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
[4] ——, "Motion Planning," *IEEE Robot. Autom. Mag.*, vol. 18, no. 1, pp. 79–89, Mar. 2011.
[5] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, 1996.
[6] S. M. LaValle, "Rapidly-Exploring Random Trees A New Tool for Path Planning," 1998.
[7] M. Elbanhawi and M. Simic, "Sampling-Based Robot Motion Planning: A Review," *IEEE Access*, vol. 2, pp. 56–77, 2014.
[8] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *Int. J. Robot. Res.*, vol. 21, no. 12, pp. 1031–1052, 2002.
[9] J. Vannoy and J. Xiao, "Real-Time Adaptive Motion Planning (RAMP) of Mobile Manipulators in Dynamic Environments With Unforeseen Changes," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1199–1212, Oct. 2008.
[10] S. M. L. Valle, "Motion Planning," *IEEE Robot. Autom. Mag.*, vol. 18, no. 2, pp. 108–118, Jun. 2011.
[11] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, Mar. 1985, pp. 500–505.
[12] D. Wang, "A generic force field method for robot real-time motion planning and coordination," PhD Thesis, 2009.
[13] A. A. A. Rizqi, A. I. Cahyadi, and T. B. Adji, "Path planning and formation control via potential function for UAV Quadrotor," in *Proc. Int. Conf. Adv. Robot. Intell. Syst.*, Jun. 2014, pp. 165–170.
[14] A. Valero-Gomez, J. V. Gomez, S. Garrido, and L. Moreno, "The Path to Efficiency: Fast Marching Method for Safer, More Efficient Mobile Robot Trajectories," *IEEE Robotics Automation Magazine*, vol. 20, no. 4, pp. 111–120, dec 2013.
[15] O. Arslan and D. E. Koditschek, "Exact robot navigation using power diagrams," in *Proc. IEEE Int. Conf. Robot. Autom.*, may 2016, pp. 1–8.
[16] L. Singh, H. Stephanou, and J. Wen, "Real-time robot motion control with circulatory fields," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, Apr. 1996, pp. 2737–2742 vol.3.
[17] L. Singh, J. Wen, and H. Stephanou, "Motion planning and dynamic control of a linked manipulator using modified magnetic fields," in *Proc. IEEE Int. Conf. Control Applicat.*, Oct. 1997, pp. 9–15.
[18] S. Haddadin, R. Belder, and A. Albu-Schäffer, "Dynamic motion planning for robots in partially unknown environments," in *IFAC World Congr.*, vol. 18, 2011.
[19] F. Zhang, E. Justh, and P. Krishnaprasad, "Boundary following using gyroscopic control," in *Proc. IEEE Conf. Decision Control*, vol. 5. IEEE, 2004, pp. 5204–5209.
[20] D. E. Chang and J. E. Marsden, "Gyroscopic forces and collision avoidance with convex obstacles," in *New trends in nonlinear dynamics and control and their applications*. Springer, 2003, pp. 145–159.
[21] A. Savkin and C. Wang, *Safe Robot Navigation Among Moving and Steady Obstacles*. Elsevier Science & Technology Books, 2015. [Online]. Available: https://books.google.co.uk/books?id=0lhorgEACAAJ
[22] G. Garimella, M. Sheckells, and M. Kobilarov, "A stabilizing gyroscopic obstacle avoidance controller for underactuated systems," in *Proc. IEEE Conf. Decision Control*, Dec. 2016, pp. 5010–5016.
[23] E. W. Justh and P. Krishnaprasad, "Equilibria and steering laws for planar formations," *Syst. & control Lett.*, vol. 52, no. 1, pp. 25–38, 2004.
[24] L. Sabattini, C. Secchi, and C. Fantuzzi, "Collision avoidance using gyroscopic forces for cooperative Lagrangian dynamical systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 953–958.
[25] ——, "Collision avoidance for multiple Lagrangian dynamical systems with gyroscopic forces," *Int. J. Adv. Rob. Syst.*, vol. 14, no. 1, p. 1729881416687109, 2017. [Online]. Available: http://dx.doi.org/10.1177/1729881416687109
[26] A. Ataka, H. K. Lam, and K. Althoefer, "Reactive Magnetic-field-inspired Navigation for Non-holonomic Mobile Robots in Unknown Environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2018, pp. 6983–6988.
[27] D. Halliday, R. Resnick, and J. Walker, *Fundamentals of Physics Extended, 10th Edition*. Wiley, 2013. [Online]. Available: https://books.google.co.uk/books?id=DTccAAAAQBAJ
[28] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, vol. 3, 2009, p. 5.